Multidisciplinary Journal for Applied Research in Engineering and Technology (MJARET)

Vol. III Issue 2, June – 2023, Pages: 01-05

DOI : https://doi.org/10.54228/mjaret0623001 **ISSN : 2583-1100**

Optimization of Graph Clustering Algorithm with Enhanced Security using Homomorphic Encryption in Block Chain

Dr. D.Jayalatchumy
Assistant Professor/Dept of CSE
Perunthalaivar Kamarajar Institute of Engineering and Technology,
Nedungadu. Karaikal. India.
djpkiet@gmail.com

P.Segar
Lab Instructor/ Dept of CSE
Perunthalaivar Kamarajar Institute of Engineering and Technology,
Nedungadu. Karaikal. India.
segarkv@gmail.com

Abstract: Power iteration clustering (PIC) is a clustering algorithm used to partition data into groups based on similarity. Blockchain technology has been widely adopted for data storage and management due to its immutability and security features. However, privacy and security concerns arise when sensitive data is stored on the blockchain. Homomorphic encryption (HE) is a cryptographic technique that can enable secure computation on encrypted data, and can be used to address these concerns. This paper proposes the implementation of PIC algorithm in blockchain with enhanced security using Homomorphic Encryption. The proposed framework ensures privacy and security of the data by using HE to encrypt and decrypt the data. This also enables secure computation on encrypted data, without the need to decrypt it first. The PIC algorithm is used for clustering the data, which ensures that similar data points are grouped together. Implementing PIC algorithm in blockchain with enhanced security using HE has the potential to address privacy and security concerns in blockchain-based systems, while enabling effective and efficient clustering of data. This can enable new use cases for blockchain technology in domains such as healthcare, finance, and e-commerce, where privacy and security of data are critical.

KEYWORDS: Homomorphic Encryption, Power Iteration Clustering, Blockchain, Privacy.

INTRODUCTION

Blockchain technology [16,17] has emerged as a secure and decentralized platform for managing transactions and data in various domains. Machine learning (ML) algorithms have also gained popularity for their ability to extract insights from large datasets. However, there are concerns about the security and privacy of sensitive data used in ML algorithms. Homomorphic encryption (HE)[11] is a cryptographic technique that can enable secure computation on encrypted data, and can be used to address these concerns. This paper proposes the implementation of ML algorithms in blockchain with enhanced security using HE. The steps involved for implementation is as follows:

- Data preprocessing: The data is preprocessed to ensure that it is in a suitable format for use in ML algorithms.
- Homomorphic encryption: The data is encrypted using HE to ensure privacy and security.

- Training the ML model: The encrypted data is used to train the ML model. This involves computing the weights and biases of the model using HE.
- Prediction: The trained ML model is used to make predictions on new data. This involves computing the output of the model using HE.
- Decryption: The predicted output is decrypted using HE to obtain the original output.

The proposed framework ensures privacy and security of the data by using HE to encrypt and decrypt the data. This also enables secure computation on encrypted data, without the need to decrypt it first. The Power Iteration Clustering algorithm[4,5] is used for training and prediction, which ensures that insights can be extracted from the data without compromising its security or privacy.. Overall, implementing PIC algorithm in blockchain with enhanced security using HE[12] has the potential to address privacy and security concerns in blockchain-based systems, while enabling effective use of PIC for extracting insights from large datasets. This can enable new use cases for blockchain technology in domains such as healthcare, finance, and e-commerce, where privacy and security of data are critical.

THE POWER ITERATION CLUSTERING (PIC)

The Power Iteration Clustering (PIC[6,7,8]) algorithm is a graph clustering algorithm that is used to cluster data points based on their similarity in a high-dimensional space. The algorithm operates on a similarity graph that is represented as an adjacency matrix, where the (i,j)-th entry represents the similarity between the i-th and j-th data points. The algorithm works by iteratively computing the principal eigenvector of the graph, which can be used to cluster the data points. The basic steps of the PIC algorithm are as follows:

- Construct a similarity graph: The first step is to construct a similarity graph based on pairwise similarities between data points. This can be done using a similarity measure such as the Gaussian kernel or the cosine similarity measure.
- Initialize the eigenvector: The eigenvector can be initialized to a random vector or a vector with equal entries.
- Perform power iterations: The algorithm then performs a sequence of power iterations on the similarity graph to obtain the principal eigenvector. In each iteration, the eigenvector is multiplied by the similarity matrix and normalized to have unit length. The algorithm can terminate after a fixed number of iterations or when the eigenvector converges.
- Cluster the data points: Once the principal eigenvector is obtained, it can be used to cluster the data points using the k-means algorithm. Specifically, the entries of the eigenvector can be used as features for clustering.

One advantage of the PIC algorithm [8,9] is that it can handle non-linearly separable data, which is a common problem in many clustering applications. Additionally, the algorithm is relatively fast and scalable, making it suitable for large datasets. However, the PIC algorithm can also have some limitations. For example, it may not perform well on graphs with disconnected components, and it can be sensitive to the choice of the similarity measure and the number of clusters. The procedure for PIC [86] is shown in Algorithm1.

Algorithm 1: Power Iteration Clustering

Input: Data points to be clustered

Output: Clustered Dataset

Step1. Input the dataset and similarity function

Step 2. Construct the similarity matrix $A \in R^{n*n}$

Step 3. Normalize the matrix W=$D^{-1}A$

Step 4. Generate the initial vector using its norm function

Step 5. Find its new vector and velocity by setting $v^{t+1} = Wv^t$ and $\delta^{t+1} \leftarrow |v^{t+1} - v^t|$

Step 6. Repeat the process until the convergence criteria is met.

Step 7. The convergence is decided when the difference between the two iterations is reduced.

Step 8. Obtain the vectors and cluster the resulting vectors using K- means algorithm

Step 9. Output the clustered results.

Multidisciplinary Journal for Applied Research in Engineering and Technology (MJARET)

Vol. III Issue 2, June – 2023, Pages: 01-05

DOI : https://doi.org/10.54228/mjaret0623001

ISSN : 2583-1100

BLOCKCHAIN

Blockchain is a distributed ledger technology that allows multiple parties to securely and transparently share data and conduct transactions without the need for a central authority. It was first introduced in 2008 with the launch of Bitcoin, a digital cryptocurrency that uses blockchain technology[16,17] to facilitate transactions without the need for a central bank. The basic idea behind blockchain is that it creates a decentralized database that is maintained and updated by a network of computers, or nodes, rather than a central authority. Each node on the network has a copy of the ledger, which contains a record of all transactions that have occurred on the network. When a new transaction is initiated, it is broadcast to all nodes on the network. The nodes work together to verify the transaction and add it to the ledger.

Once a transaction has been added to the ledger, it cannot be altered or deleted, providing an immutable and transparent record of all transactions on the network. Blockchain has a number of key benefits that make it ideal for a wide range of applications. For example, it provides a high level of security and transparency, as well as a decentralized and distributed infrastructure that makes it resistant to cyber-attacks and other types of malicious activity. Some of the most promising applications of blockchain technology include supply chain management, voting systems, financial transactions, and identity verification. As the technology continues to evolve and mature, it is likely that we will see many more innovative and transformative applications emerge in the years ahead.

3.1 WORKING OF BLOCKCHAIN

The core components of a blockchain include blocks, transactions, and a consensus mechanism. Each block contains a set of transactions that have been verified by the network, and is linked to the previous block, forming a chain of blocks. When a new transaction is initiated, it is broadcast to the network of nodes, who then validate the transaction using a consensus mechanism. This mechanism ensures that the transaction is valid and that the blockchain remains secure and tamper-proof. Once a transaction is validated, it is included in a block, which is then added to the blockchain. The block is then broadcast to all nodes on the network, who validate the block before adding it to their own copy of the blockchain [16].

Each block in the blockchain contains a unique cryptographic hash, which links it to the previous block in the chain. This creates an unbreakable chain of blocks, where any attempt to modify a previous block will be immediately detected and rejected by the network. This makes blockchain an extremely secure and reliable way to store and verify transaction data, as it removes the need for a centralized authority to validate and verify transactions. Instead, the network of nodes operates on a distributed consensus mechanism, ensuring that all transactions are validated and recorded in a secure and transparent manner. The working of blockchain can be described in several steps:

- Data Entry: A user or participant initiates a transaction by creating a block of data, which includes information about the transaction, such as the parties involved, the amount of currency or asset exchanged, and a timestamp.
- Verification: The block of data is verified by a network of computers called nodes, which check the data against a set of predetermined rules and algorithms to ensure that it is valid.
- Addition to Blockchain: Once the block of data is verified, it is added to a chain of previous blocks, creating a permanent and unchangeable record of the transaction. This chain is maintained by all the nodes in the network, creating a distributed ledger that is accessible to all participants.
- Consensus: In order to maintain the integrity of the blockchain, a consensus mechanism is used to ensure that all nodes in the network agree on the state of the ledger. This can be achieved through a variety of mechanisms, such as proof-of-work or proof-of-stake algorithms.
- Security: Each block in the blockchain is secured using cryptography, which makes it virtually impossible for anyone to alter the data stored in the block without being detected. This makes the blockchain an ideal solution for use cases where trust and transparency are critical.

Overall, the working of blockchain [17] is based on a set of rules and algorithms that ensure the integrity, security, and transparency of the data stored in the blockchain. By using this technology, organizations can create a secure and decentralized system for recording and verifying transactions, managing supply

chains, and much more. The procedure for implementing Power Iteration Clustering (PIC) in blockchain is given below in Algorithm 2.

Algorithm 2: PIC in Blockchain

- The data to be clustered is preprocessed to ensure that it is in a suitable format for clustering using PIC.
- The data is encrypted using homomorphic encryption (HE) to ensure privacy and security.
- The algorithm is initialized by selecting a number of initial cluster centroids from the data.
- The power iteration method is used to compute the similarity between the data points and the cluster centroids.
  - Compute the distance between each data point and each cluster centroid using HE.
  - Update the cluster assignments for each data point based on the closest centroid.
  - Recompute the centroids based on the updated cluster assignments.
  - Repeat above steps until convergence is achieved.
- Decryption: The clustered data is decrypted using HE to obtain the original data.
- Verification: The validity of the clustering results is verified by comparing them with the expected results.

4. HOMOMORPHIC ENCRYPTION

Homomorphic encryption[13] is a technique that allows computations to be performed on encrypted data without decrypting it first. This means that sensitive data can remain encrypted while still being used for important computations. With homomorphic encryption, the security of blockchain technology can be enhanced by encrypting transaction data and computations in a way that makes it impossible for anyone to view the data or the calculations performed on it. Using homomorphic encryption in a blockchain involves encrypting the data in the blockchain so that it can be securely stored and processed without revealing its contents. This can be done by using a homomorphic encryption scheme to encrypt the data before it is added to the blockchain. The encrypted data can then be stored on the blockchain and processed by authorized parties using homomorphic operations, which allow the parties to perform computations on the encrypted data without decrypting it.

One possible application of homomorphic encryption in blockchain is in secure data sharing. For example, a healthcare blockchain can store encrypted patient data, which can be accessed by authorized parties who can perform computations on the encrypted data using homomorphic operations to generate insights without compromising the privacy of the patients. Another application of homomorphic encryption in blockchain is in secure smart contracts. Smart contracts can be executed on encrypted data using homomorphic operations, which can ensure that the execution of the contract is secure and private. However, it is important to note that using homomorphic encryption in blockchain can be computationally expensive, and may require specialized hardware and software to perform efficiently [13,14,15]. Additionally, the security and privacy of the system can be compromised if the homomorphic encryption scheme is not designed and implemented correctly. The Algorithm 3 explains the steps for Homomorphic Encryption in PIC

Input: Data points to be clustered
Output: Clustered dataset

Step 1: Input the data points to be checked.
Step2: Apply Power Iteration Clustering Algorithm(PIC) to the matrix representation data points to identify the clusters (Refer Algo 1)
Step 3: Repeat step2 until convergent
Step 4: Obtain the clusters from step 3 each represent by centroid vector
Step 5: Use any of the Homomorphic Encryption scheme to encrypt the centroid
Step 6: Perform computation on encrypted centroid vectors without revealing its data points
Step 7: Apply the inverse of PIC algorithm to decrypt the centroid vectors
Step 8: Decrypt the results of computation to obtain the clustered data.

Multidisciplinary Journal for Applied Research in Engineering and Technology (MJARET)

Vol. III Issue 2, June – 2023, Pages: 01-05

DOI : https://doi.org/10.54228/mjaret0623001 **ISSN : 2583-1100**

IMPLEMENTATION RESULTS

Sample Code for PIC in Blockchain

```
    # Choose a random vector to start with
    b_k = np.random.rand(M.shape[0])
    for _ in range(max_iterations):
        # Calculate the matrix-by-vector product Mb
        b_k1 = np.dot(M, b_k)
        # Calculate the norm
        b_k1_norm = np.linalg.norm(b_k1)
        # Re-normalize the vector
        b_k = b_k1 / b_k1_norm
    return b_k

def pic(data, num_clusters, max_iterations):
    # Compute the similarity matrix
    S = np.dot(data, data.T)
    # Apply the power iteration method
```

Sample code for HE PIC in blockchain

```
def dot_product(vec1, vec2, public_key):
    dot_prod = public_key.encrypt(0)
    for i in range(len(vec1)):
        prod = vec1[i] * vec2[i]
        dot_prod += prod
    return dot_prod

def he_pic_algorithm(adjacency_matrix, num_clusters,
    # Encrypt the adjacency matrix
    encrypted_matrix = encrypt_matrix(adjacency_matri

    # Initialize a random vector
    v = [public_key.encrypt(random.uniform(0, 1)) for

    # Run power iteration
    for i in range(100):
```

Results

```
Centroids: [[0.39585426 0.4977648  0.64498749 0.51727
 0.48115378 0.41203667 0.5837168  0.41762035]
[0.41923631 0.54003917 0.48771214 0.41750817 0.42736
 0.38808903 0.47312938 0.48147467 0.53773189]
[0.41983729 0.49015671 0.51246596 0.44877508 0.58353
 0.42811444 0.55601375 0.47584381 0.51800377]
[0.5202769  0.54929753 0.45334136 0.54762468 0.49944
 0.50019615 0.56602112 0.41020105 0.51872891]
[0.46544344 0.49821716 0.41933868 0.54339419 0.48494
 0.51585126 0.54749717 0.57682054 0.46285491]]
```

CONCLUSION

The implementation of machine learning algorithms in blockchain with enhanced security using homomorphic encryption has several advantages. It enables secure and private computation on sensitive data stored on the blockchain, making it an ideal solution for applications that require privacy and security. The use of homomorphic encryption ensures that the data remains encrypted throughout the computation, and the results are only decrypted by the authorized parties. This protects the data from unauthorized access and ensures privacy and security. Moreover, the integration of machine learning algorithms in blockchain can help automate decision-making processes and improve the efficiency of various applications. This can help reduce costs and improve the quality of services provided. Overall, the use of homomorphic encryption and machine learning algorithms in blockchain has the potential to revolutionize the way data is processed, stored, and analyzed, providing new opportunities for innovation and growth. However, further research is needed to optimize the performance of these techniques and address the challenges associated with their implementation.

REFERENCES

[1] Dartmouth, Numerical methods. Chapter 10.3 Power method for approximating eigen values, ENGS 91, Dartmouth College Hanover. New Hampshire.

[2] Dhanapal Jayalatchumy, Perumal Thambidurai. 2016, Inflated Power Iteration Clustering Algorithm to Optimize Convergence Using Lagrangian Constraint. Advances in Intelligent system and computing, Proceedings of 5th Computer Science Online Conference (CSOC 16)(2) pp. 227-238

[3] Frank Lin frank, W. William, 2010. Power Iteration Clustering, International Conference on Machine Learning, Haifa, Israel

[4] Chen, W.Y., Y. Song, H. Bai, C. Lin, E.Y. Chang, 2011. Parallel spectral clustering in distributed systems, IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(3): 568-586.

[5] Weizhong Yana, 2013. p-PIC: Parallel power iteration clustering for Big Data, Models and Algorithms for High Performance Distributed Data Mining, 73-3.

[6] Jayalatchumy Dhanpal, Thambidurai Perumal, and Udhayakumaran. 2016. Efficient Graph Clustering Algorithm and its use in Prediction of Students Performance. In Proceedings of the International Conference on Informatics and Analytics (ICIA-16). Association for Computing Machinery, New York, NY, USA, Article 39, 1–5. https://doi.org/10.1145/2980258.2980343

[7] Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: Advances in Neural Information Processing Systems, vol. 14, pp. 849–856. MIT Press, Cambridge (2002)

[8] The, A.P., Thang, N.D., Vinh, L.T., Lee, Y.-K., Lee, S.: Deflation-based power iteration clustering. Appl. Intell. 39(2), 367–385 (2013)

[9] Yan, W., et al.: p-PIC: parallel power iteration clustering for big data. J. Parallel Distrib. Comput. 73(3), 352–359 (2013)

[10] Dartmouth: Numerical methods. Chapter 10.3 Power method for approximating eigenvalues, ENGS 91, Dartmouth College, Hanover, New Hampshire

[11] F. O. Catak, I. Aydin, O. Elezaj, and S. Yildirim-Yayilgan, "Practical Implementation of Privacy Preserving Clustering Methods Using a Partially HomomorphicEncryption lgorithm," Electronics, vol. 9, no. 2, p. 229, Jan. 2020, doi: 10.3390/electronics9020229.

[12] A Vijaya Kumar, Mogalapalli Sai Sujith et.al " Secure Multiparty computation enabled E-Healthcare system with Homomorphic encryption", December 2020, IOP Conference Series Materials Science and Engineering 981(2):022079, DOI: 10.1088/1757-899X/981/2/022079

[13] Munjal, K., Bhatia, R. A systematic review of homomorphic encryption and its contributions in healthcare industry. Complex Intell. Syst. (2022). https://doi.org/10.1007/s40747-022-00756-z

[14] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. 2018. A Survey on Homomorphic Encryption Schemes: Theory and Implementation. ACM Comput. Surv. 51, 4, Article 79 (July 2019), 35 pages. https://doi.org/10.1145/3214303

[15] Hamza R, Hassan A, Ali A, Bashir MB, Alqhtani SM, Tawfeeg TM, Yousif A. Towards Secure Big Data Analysis via Fully Homomorphic Encryption Algorithms. Entropy. 2022; 24(4):519. https://doi.org/10.3390/e24040519

[16] Purwono Purwono,, Blockchain Technology, Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)Vol. 8, No. 2, June 2022, pp. 199-205ISSN: 2338-3070.

[17] Neil Efren Villanueva, Blockchain Technology Application: Challenges, Limitations and Issues, Journal of Computational Innovations and Engineering Applications 5(2) 2021: 8–14